Python Fun(damentals)

Python Types

Alexander Rymdeko-Harvey

Obscurity Labs

```
* Innovation
* Expert Training
* Advanced Security Services
```

Copyright (c) 2018 Alexander Rymdeko-Harvey & Obscurity Labs LLC.

Introduction

- Who am I?
- Why this is important?
- Free training?

Install Demo and Instructions

1. Windows/Mac/Ubuntu Install Instructions
2. Python 3.6+ & Packages
3. Visual Studio Code
4. Pipenv Instruction

Python 3 Types

Its important to understand C/C++ principles first, to understand what makes Python types so easy to work with. Lets take the following example in C:

```c
/* variable definition: */
int a = 1;
char b ='G';
double c = 3.14;
```

Now in Python, notice we dont declare it statically?

```python
# variable definition
a = 1
b ='G'
c = 3.14
```

Python 3 Types cont.

Five standard types you need know:

- Numeric Types - int, float, complex
- Text Sequence Type - str
- Sequence Types - list, tuple, range
- Set Types - set, frozenset
- Mapping Types - dict
- Binary Sequence Types — bytes, bytearray, memoryview

Python Numberic Types

- There are three distinct numeric types: integers, floating point numbers, and complex numbers
- Booleans are a subtype of integers
- Integers have unlimited precision
- Floating point numbers are usually implemented using double in C

Python Numberic Types Cont.

Operation	Result
x + y	sum of x and y
x - y	difference of x and y
x * y	product of x and y
x / y	quotient of x and y
int(x)	x converted to integer
float(x)	x converted to floating point
pow(x, y)	x to the power y
x ** y	x to the power y

Python Numeric Built-in Methods

Many of the `Types` within Python have methods and in your IDE such as VSC will allow you to explore these.

Here is a example of `int.bit_length()`:

```
>>> n = -37
>>> bin(n)
'-0b100101'
>>> n.bit_length()
6
```

NOTE: bin() is the binary representation

Here is a example of `int.to_bytes()`:

```
>>> (1024).to_bytes(2, byteorder='big')
b'\x04\x00'
```

https://docs.python.org/3/library/stdtypes.html#int.bit_length

https://docs.python.org/3/library/stdtypes.html#int.to_bytes

Lab_1.py

TASKING

Perfrom the following on the variable `dataNum`:

1. Set Value to `1.2299`
2. Set `dataNumPower` to power of 2 for `dataNum`
3. Set `dataNum2` to the int `10`
4. Set `dataNum2Bytes` to the bytes of `dataNum2`, setting the length to `1` and the byte order to `big`

Python Text Sequence Type

- Textual data in Python is handled with str objects, or strings
- Strings are immutable sequences of Unicode code points.
- String literals are written in a variety of ways
  -Single quotes: 'allows embedded "double" quotes'
  -Double quotes: "allows embedded 'single' quotes".
  -Triple quoted: '''Three single quotes''', """Three double quotes"""
- Triple quoted strings may span multiple lines

https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str

Python Text Sequence Operator Cont.

String `+` Operator

```
>>> a = 'alex' + 'rymdekeo-harvey'
>>> b = 'alex' + ' rymdeko-harvey'
>>> print(a)
alexrymdekeo-harvey
>>> print(b)
alex rymdeko-harvey
```

String `in` Operator

```
>>> 'alex' in 'alex rymdeko-harvey'
True
```

Python Text Sequence Operator Cont.

String Indexing

```
>>> a = 'ALEX'
>>> print(a[1])
L
```

What does this look like in C/C++?

```
// valid initialization
char name[10] = {'A','L','E', 'X','\0'};
```

What does this look like in the Python Object?

A	L	E	X
0	1	2	3

Python Text Sequence Methods

- Strings implement all of the common sequence operations
- With the additional methods
- Python provides TONS of string methods we will only cover a few but labs may require research
- Most common style is Format String Syntax

Python Text Sequence Methods Cont.

Here is a example of `str.capitalize()`:

```
# Return a copy of the string with its first
# character capitalized and the rest lowercased.
>>> a = 'alex'
>>> a.capitalize()
'Alex'
```

Here is a example of `str.capitalize()`:

```
# Return a copy of the string with its first
# character capitalized and eading characters removed.
>>> a = '  alex  '
>>> a.lstrip().capitalize()
'Alex  '
```

https://docs.python.org/3/library/stdtypes.html#str.capitalize

https://docs.python.org/3/library/stdtypes.html#str.lstrip

Lab_2.py

TASKING

Perfrom the following on the variable `dataStr`:

1. Set `dataStr` value to 'opensource.com'
2. Set `dataStrFull` full value of `https://` and use the `dataStr` to create this variable
3. Set `dataStrSplit` to the value of `dataStrFull` split on the the character `.`, into a list using a string method (*HINT: Python Docs*)
4. Try to print `dataStrSplit` with the format string methods like `print('{} {}'.format('one', 'two'))`